

# HTML 5 ja JavaScript API:t

HTML5 toi useita merkittäviäkin ohjelmointirajapintoja (API) käytettäväksi. Tässä luvussa esitellään niistä merkittävimpiä. Osaa näistä tekniikoista on sovellettava harjoitustehtävissä ja osaamistestissä. Osaa voit hyödyntää harjoitustyössäsi.

## Canvas

- Canvas-elementti on yksinkertainen suorakulmainen reunaviivaton alue grafiikan ja kuvien esittämiseen
- Canvas-elementti saa merkityksensä vasta sitten kun siihen piirretään JavaScriptillä
- Merkittävintä canvasissa on se, että siihen voidaan tuottaa kuvia **dynaamisesti**, jolloin
  - muuttuvaisen kuvan voi tuottaa esim palvelinohjelman syöttämän datan perusteella (esim. esitysgrafiikkaa: säätilastoja, pörssikursseja yms.)
  - muuttuvainen kuva voidaan toteuttaa käyttäjän syötteiden perusteella (esim. selaimessa toimiva piirto-ohjelma yms.)

### Esimerkki 0701

Piirretään vaaleansininen suorakaiteen muotoinen alue

```
<canvas id="canvas" width="600" height="400"></canvas>
```

Liitetään elementti JavaScriptillä käsiteltäväksi:

```
var canvas = document.getElementById ("canvas");  
var context = canvas.getContext("2d");
```

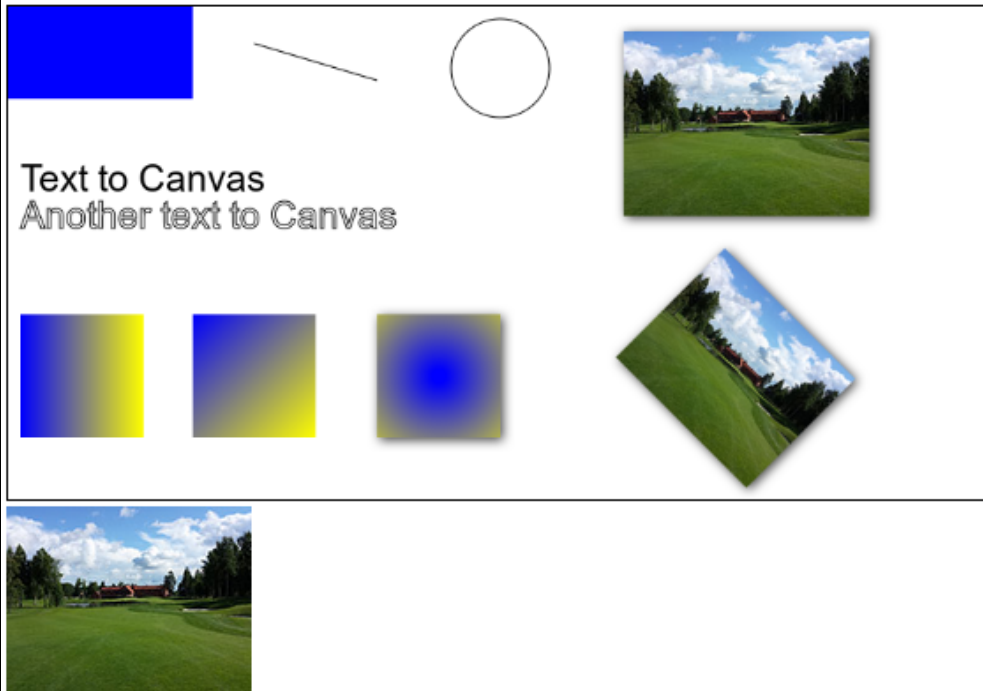
Alueen täyttö:

```
context.fillStyle = "lightblue";  
context.fillRect(50,50,150,150);
```

### Esimerkki 0702

Peruspiirtelyä Canvasiin

## My Canvas



## Lähdekoodi

Tutki ja kokeile

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Canvas Examples</title>
<script>
function drawToCanvas() {
    // find the <canvas> element
    var canvas = document.getElementById("myCanvas");
    // get canvas context
    var ctx = canvas.getContext("2d");
    // define fill style
    ctx.fillStyle="#0000FF";
    // draw filled rectangle - fillRect(x,y,width,height)
    ctx.fillRect(0,0,150,75);
    // draw a path
    ctx.moveTo(200,30); // from
    ctx.lineTo(300,60); // to
    ctx.stroke();        // line
    // circle
    ctx.beginPath();
    // x, y, r, start, stop
    ctx.arc(400,50,40,0,2*Math.PI);
    ctx.stroke();
    // text
    ctx.fillStyle="#000000";
    ctx.font="30px Arial";
    ctx.fillText("Text to Canvas",10,150);
    ctx.strokeText("Another text to Canvas",10,180);

    // gradients
    // create linear gradient - createLinearGradient(x,y,x1,y1)
    // left to right
    var grd=ctx.createLinearGradient(10,250,110,250);
    grd.addColorStop(0,"blue");
    grd.addColorStop(1,"yellow");
    // Fill with gradient
    ctx.fillStyle=grd;
    // TopLeftCorner X, Y, width, height
    ctx.fillRect(10,250,100,100);
    // another - left top to right bottom
    var grd2=ctx.createLinearGradient(150,250,250,350);
    grd2.addColorStop(0,"blue");
    grd2.addColorStop(1,"yellow");
    ctx.fillStyle=grd2;
    ctx.fillRect(150,250,100,100);
}
```

```

        grad2.addColorStop(0,"blue");
        grad2.addColorStop(1,"yellow");
        // Fill with gradient
        ctx.fillStyle=grad2;
        ctx.fillRect(150,250,100,100);

        // create radial gradient - createRadialGradient(x,y,r,x1,y1,r1)
        // first three are "inner circle" others "outer circle"
        // center is 350,300 (in following fillrect)
        var grd3=ctx.createRadialGradient(350,300,5,350,300,100);
        grd3.addColorStop(0,"blue");
        grd3.addColorStop(1,"yellow");
        // Fill with gradient with shadow
        ctx.fillStyle=grd3;
        ctx.shadowOffsetX = 2;
        ctx.shadowOffsetY = 2;
        ctx.shadowBlur = 8;
        ctx.shadowColor = "rgba(0, 0, 0, 0.75)";
        ctx.fillRect(300,250,100,100);

        // image
        var image = document.getElementById("image");
        ctx.drawImage(image,500,20);

        // translate center of canvas to (-200,-80) of bottom right corner
        ctx.translate(canvas.width-200,canvas.height-80);
        // rotate 45 degrees clockwise
        ctx.rotate(Math.PI/4);
        // draw image
        ctx.drawImage(image,image.width/-2,image.height/-2,image.width-50,image.height-25);
    }
</script>
</head>

<body onload="drawToCanvas()">
<div>
    <h2>My Canvas</h2>
    <canvas id="myCanvas" width="800" height="400" style="border:1px solid #000000;"/>
</div>
<div>

</div>

</body>
</html>

```

## Animaatiot

- JavaScript-animoinnin välineet ovat melko karuja
  - DOM-elementtejä piirrellään sivulla uusiin sijainteihin
  - Elementtejä tulee piirrellä annetussa aika-jaksossa tai kuvanopeuden (frame rate) mukaan.
  - JavaScriptissä ei ole pause/sleep-toimintoja
  - Käytä setTimeout- ta setInterval-funktioita
    - setTimeout(fCall,100) // funktiokutsu suoritetaan kerran 100msec kuluttua
    - setInterval (fCall,100) // toistuvasti kunnes lopetetaan

### Esimerkki 0703

```

function move() {
    div.style.left = parseInt(div.style.left)+1+"px";
    // recursive call every 17 millisecs
    setTimeout(move,17);
}

function init() {
    div = document.getElementById("myDiv");
    div.style.left = "0px";
    move();
}

<body onload="init();">
<div id="myDiv">Sample DIV</div>

```

- Samaa tekniikkaa voidaan käyttää canvasin kanssa
- Älä siirrä DOM-elementtejä, piirrä objekti uuteen sijaintiin

## Animaation sulavuus

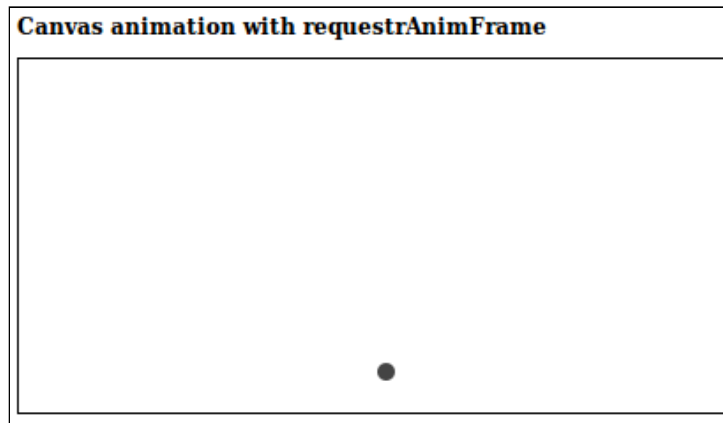
- Animaation sulavuus riippuu animaation frame ratesta
- Tavallisesti filmit 24fps, videot 30fps
- Näytön virkistystaajuus 60Hz -> enempää ei kannata tavoitella
  - $1s = 1000ms$
  - $1000ms/60(fps) = 16.7ms$
  - `setInterval(teeJotain, 17)`

## requestAnimationFrame

- `setTimeout` ja `setInterval` eivät ole ongelmattomia
  - eivät huomioi muita selaintapahtumia
- Mozilla esitteli `requestAnimationFrame`-funktion (tarjolla myös WebKitissä: Chrome, Safari)
- Natiivi API kaikille selainanimaatioille
- <https://developer.mozilla.org/en-US/docs/Web/API/window.requestAnimationFrame>

## Esimerkki 0704 - Canvas-animaatio

- Pallon liikuttelu canvas-elementissä
- Tutustu toimintaan ja lähdekoodeihin
  - `setInterval`: [03-canvas-animate.html](#)
  - `requestAnimationFrame`: [03-canvas-animate-requestAnimationFrame.html](#)



## Esimerkki 0704 - Breakout Game

- Objektit: Breakout, Ball, Paddle ja Block
- Canvas ja requestAnimationFrame
- Törmäykset: Perusmatematiikkaa

[04-breakout-game](#)



- API käyttäjän maantieteellisen sijainnin määrittelyä
- Sijainti selvitetään: GPS:llä, IP-osoitteen tai matkapuhelinverkon tukiaseman perusteella
- Yhdistettynä vaikkapa Googlen karttoihin käyttäjän sijainti helppo näyttää
- Toimii ainoastaan **https**-yhteyttä käytettäessä!

### Esimerkki 0705 -Sijainnin näyttäminen



Esimerkki EI toiminnassa: [esimerkit/11/05-geolocation/05-geolocation-basic.html](https://esimerkit/11/05-geolocation/05-geolocation-basic.html)

Esimerkki ok, mutta vaatisi https:n toimiakseen!

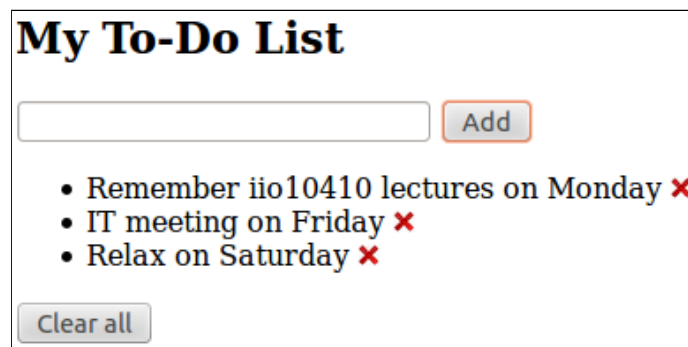
---

## HTML5 Local and Session Storage

- Aiemmin web-sovellukset ovat pystyneet tallentamaan dataa asiakaspäähän hyvin rajoitetusti (palvelimen asettamat evästeet)
- Local ja Session Storage tarjoavat uuden paikan ja tavan päästä asiakaspäähän tallennettuun dataan käsiksi JavaScriptillä
- arkaluonteista dataa ei kannata tallentaa localStorageen
- data on tallennettu nimi-arvo-pareina ja web-sivu pääsee käsiksi ainoastaan itse tallentamaansa dataan
- **localStorage**: data säilyy vaikka selaimen sulkee (no expiration date)
- **sessionStorage**: data ei säily jos selaimen sulkee (data säilyy selainistunnon ajan)

### Esimerkki 0706 - Local Storage

- Lisää kohteita Local Storageen ja UL-elementtiin
- Näytä tallennetut kohteet, kun selain käynnistetään uudelleen (ja ladataan ao. sivu)
- Poista kohteita Local Storagesta ja UL-elementistä



The screenshot shows a web application titled "My To-Do List". It features a text input field at the top with an "Add" button to its right. Below the input field, there is a list of three items: "Remember iio10410 lectures on Monday", "IT meeting on Friday", and "Relax on Saturday". Each item is followed by a red "X" icon. At the bottom of the list, there is a "Clear all" button.

Lähdekoodit: [07-local-storage.html](https://github.com/07-local-storage.html)

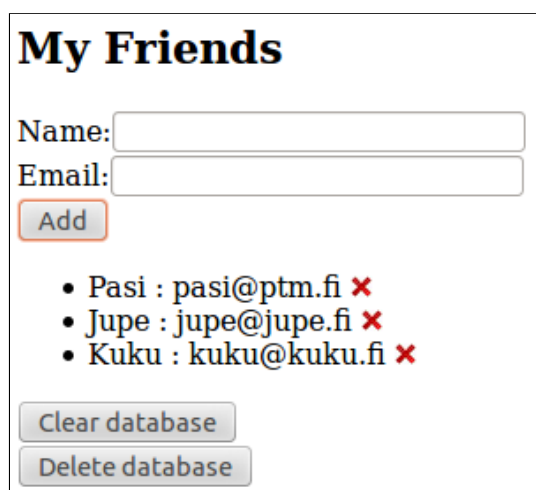
---

## Indexed Database API

- Objektipohjainen datavarasto (ei tauluja, rivejä ja sarakkeita)
- Isoilla datamäärillä tehokkaampi kuin localStorage
- Ei niin yksinkertainen kuin local ja sessionStorage

### Esimerkki 0707 - My Friends

- luo tietokanta ystäviä varten
- lisää ystävä (id, name, email)
- Tyhjä ja poista tietokanta



The screenshot shows a web application titled "My Friends". It has a form with two input fields: "Name:" and "Email:". Below these fields is an "Add" button. Under the "Add" button, there is a list of three items: "Pasi : pasi@ptm.fi", "Jupe : jupe@jupe.fi", and "Kuku : kuku@kuku.fi". Each item is followed by a red "X" icon. At the bottom of the list, there are two buttons: "Clear database" and "Delete database".

Lähdekoodit: [08-indexed-db.html](https://github.com/08-indexed-db.html)

---

## Web Workers


- JavaScript on yksisäikeinen ympäristö -> samaan aikaan ei voi ajaa useita skriptejä

- Web Workers tuo säkeistuksen JavaScriptiin
- Web Workers on API taustaskriptien käynnistämiseen
- käyttötapauksia: kuvien prosessointi, tausta I/O, laajojen JSON- tai XML-vastausten prosessointi, tietokannan päivittäminen jne...

## Esimerkki 0708

Raskaan laskennan suorittaminen samanaikaisesti kartan siirtelyn kanssa

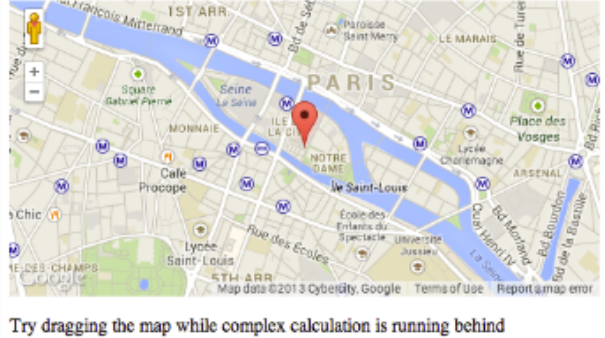
### Web Workers Example



Try dragging the map while complex calculation is running behind

**Map and UI is jammed during calculation**

### Web Workers Example



Try dragging the map while complex calculation is running behind

Complex calculation ready with workers - result = 1000000000

**Map and UI is usable during calculation**

Lähdekoodit: [09-web-workers.html](http://09-web-workers.html)

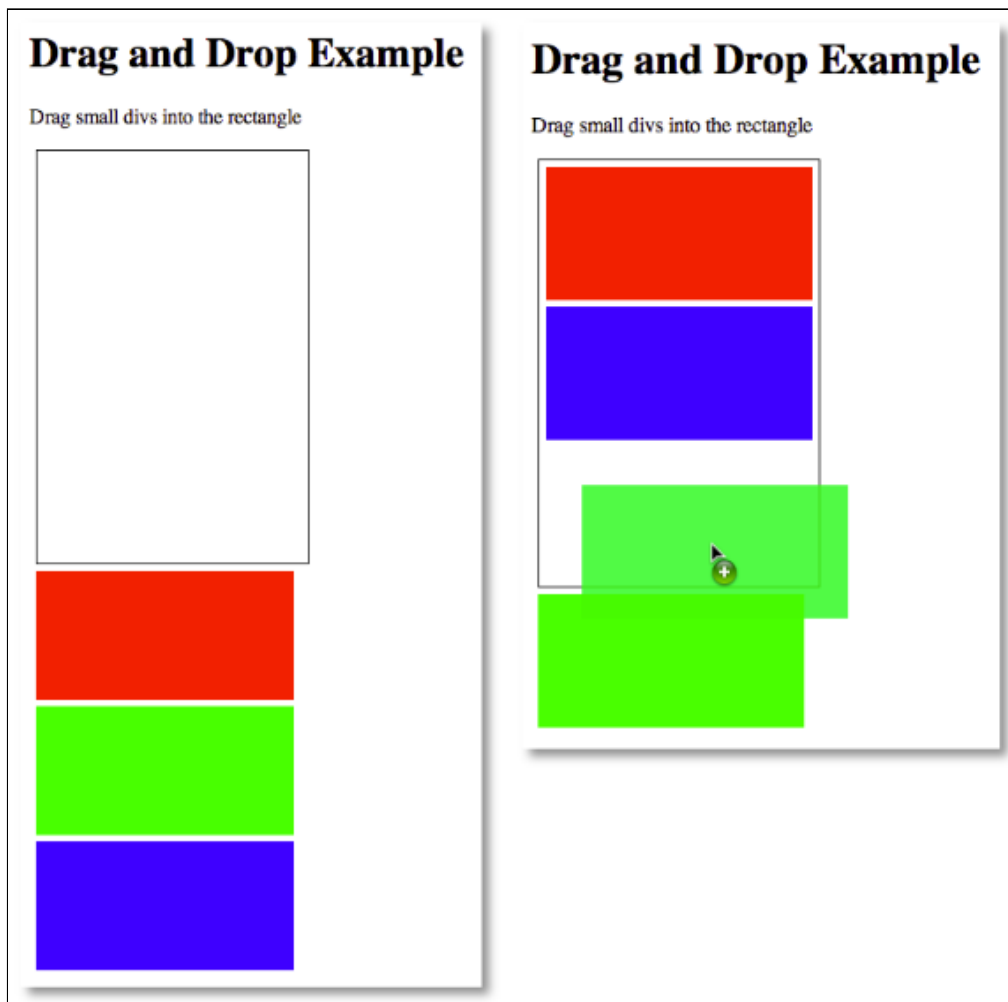
## Drag and Drop

- HTML5:n osa (Kaikki elementit raahattavia)
- saatavilla yhdelle web-sivulle tai web-sivustojen välillä käytettäväksi
- jonkin verran monimutkainen API -> jQueryn Drag and Drop paljon helpompi
- Tapahtumia
  - ondragstart
  - ondragover
  - ondrop
- toteutettu tiedonsiirto-objektina (DataTransfer object)
  - objekti pitää datan vedä-pudota-toimenpiteen ajan
  - saattaa pitää yhden tai useamman data-kohteen.

## Esimerkki: Drag and Drop

- Raahataan värillisiä DIV-elementtejä suorakulmioon





HTML:

```
<div id="divToDrop" ondrop="drop(event)" ondragover="allowDrop(event)"></div>  
<div id="drag1" draggable="true" ondragstart="drag(event)"></div>
```

JavaScript:

```
function allowDrop(event) {  
    event.preventDefault();  
}  
  
function drag(event) {  
    event.dataTransfer.setData("DraggedDIV",event.target.id);  
}  
  
function drop(event) {  
    event.preventDefault();  
    var data = event.dataTransfer.getData("DraggedDIV");  
    event.target.appendChild(document.getElementById(data));  
    document.getElementById(data).draggable = false;  
}
```

Esimerkki toiminnassa: [esimerkit/11/06-drag-and-drop-basic](https://esimerkit/11/06-drag-and-drop-basic)

Jätetty tarkoituksella tyhjäksi.